

# Hybrid of Memory and Prediction Strategies for Dynamic Multiobjective Optimization

Zhengping Liang<sup>a</sup>, Shunxiang Zheng<sup>a</sup>, Zexuan Zhu<sup>a,b,\*</sup>, and Shengxiang Yang<sup>c</sup>

*a. College of Computer Science and Software Engineering, Shenzhen University, China*

*b. Shenzhen Cyberspace Laboratory, Shenzhen 518052, China*

*c. The Centre for Computational Intelligence (CCI), School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K.*

---

## Abstract

Dynamic multiobjective optimization problems (DMOPs) are characterized by a time-variant Pareto optimal front (PF) and/or Pareto optimal set (PS). To handle DMOPs, an algorithm should be able to track the movement of the PF/PS over time efficiently. In this paper, a novel dynamic multiobjective evolutionary algorithm (DMOEA) is proposed for solving DMOPs, which includes a hybrid of memory and prediction strategies (HMPS) and the multiobjective evolutionary algorithm based on decomposition (MOEA/D). In particular, the resultant algorithm (MOEA/D-HMPS) detects environmental changes and identifies the similarity of a change to the historical changes, based on which two different response strategies are applied. If a detected change is dissimilar to any historical changes, a differential prediction based on the previous two consecutive population centers is utilized to relocate the population individuals in the new environment; otherwise, a memory-based technique devised to predict the new locations of the population members is applied. Both response mechanisms mix a portion of existing solutions with randomly generated solutions to alleviate the effect of prediction errors caused by sharp or irregular changes. MOEA/D-HMPS was tested on 14 benchmark problems and compared with state-of-the-art DMOEAs. The experimental results demonstrate the efficiency of MOEA/D-HMPS in solving various DMOPs.

**Keywords:** Dynamic multiobjective optimization, Evolutionary algorithms, Similar environment, Memory, Prediction.

---

## 1. Introduction

Multiobjective optimization problems (MOPs) feature multiple conflicting objectives and are ubiquitous in the real world [8, 7]. Dynamic MOPs (DMOPs), where the number of objectives [5], decision variables [22], objective functions [46, 19, 12], and/or constraints [42] vary over time, form a subclass of  
5 MOPs. They can be defined in various mathematical forms according to the different types of dynamic

---

\*Corresponding author

Email address: liangzp@szu.edu.cn; zsx0229@qq.com; zhuzx@szu.edu.cn; syang@dmu.ac.uk (Zhengping Liang<sup>a</sup>, Shunxiang Zheng<sup>a</sup>, Zexuan Zhu<sup>a,b,\*</sup>, and Shengxiang Yang<sup>c</sup>)

changes involved [34, 11]. A general DMOP can be defined as

$$\begin{aligned}
\min \quad & F(x, t) = (f_1(x, t), \dots, f_{m_t}(x, t))^T \\
\text{s.t.} \quad & g_i(x, t) \leq 0, i = 1, 2, \dots, p \\
& h_j(x, t) = 0, j = 1, 2, \dots, q \\
& x \in \Omega_x, \quad t \in \Omega_t
\end{aligned} \tag{1}$$

where  $x = (x_1, x_2, \dots, x_{n_t})^T$  represents an  $n_t$ -dimensional decision variable vector,  $t$  refers to a discrete time instance, and  $m_t$  indicates the number of objectives. The values of  $m_t$  and  $n_t$  may change over time.  $F(x, t) \in R^{m_t}$  denotes the objective function vector.  $p$  and  $q$  are the numbers of inequality and equality constraints, respectively.  $g_i(x, t)$  represents the  $i$ -th inequality constraint and  $h_j(x, t)$  the  $j$ -th equality constraint.  $\Omega_x \in R^{n_t}$  and  $\Omega_t$  denote the feasible decision space and time space, respectively.

Let  $x$  and  $y$  be two candidate solutions of a target DMOP, as defined in (1).  $x$  is said to dominate  $y$  at time step  $t$ , denoted by  $x \prec_t y$ , if and only if  $\forall i \in \{1, 2, \dots, m\}, f_i(x, t) \leq f_i(y, t)$  and  $\exists j \in \{1, 2, \dots, m\}, f_j(x, t) < f_j(y, t)$ . If a solution  $x^*$  is not dominated by any other solution,  $x^*$  is called a Pareto-optimal solution (non-dominated solution). All the Pareto-optimal solutions at time step  $t$  form the  $t$ -th Pareto optimal set (PS), denoted by  $PS_t = \{x^* | \nexists x \in \Omega_x, x \prec_t x^*\}$ . The corresponding objective vector set of  $PS_t$  is known as the Pareto front (PF) at time step  $t$ , i.e.,  $PF_t = \{F(x^*, t) | x^* \in PS_t\}$ . According to the different dynamic characteristics of the PS and PF, DMOPs can be divided into four types [11]: 1) the PS changes over time, but the PF is fixed; 2) both the PS and PF change over time; 3) the PS is fixed, whereas the PF changes over time; and 4) both the PS and PF are fixed, but the problem changes. Because very few instances of the last type have been studied in depth, this study was focused on the first three types of DMOPs.

Multiobjective optimization evolutionary algorithms (MOEAs) have been widely used to solve DMOPs [46, 33, 38, 17] by virtue of their ability to find diverse tradeoff solutions and approximate the PS efficiently in a single run [24, 43, 9, 44, 45]. However, the changes in the PF and/or PS in DMOPs still pose significant challenges to traditional MOEAs. Dynamic MOEAs (DMOEAs) were further proposed to track a moving PF/PS quickly and obtain PSs that are uniformly distributed over time. Most of the existing DMOEAs are constructed by combining classical MOEAs with effective dynamic handling techniques, including prediction-based [17, 47, 26], memory-based [21, 31], and diversity-based methods [10, 1, 2]. These techniques have shown competitive performances for tackling DMOPs, but each is usually limited to solving a specific group of DMOPs. In particular, prediction-based methods are specialized for solving DMOPs with predictable dynamic changes, memory-based methods are dedicated to handling DMOPs with cyclic changes, and diversity-based methods are tailored to address DMOPs having dynamic changes that can cause serious loss of diversity. A more desirable DMOEA is expected to solve DMOPs with various difficulties effectively.

It is noted that the dynamic changes of many DMOPs are both predictable and cyclic. Accordingly, hybrid methods based on memory and prediction response strategies were proposed for solving DMOPs with these two types of changes [21, 27]. In this paper, we introduce a hybrid of memory and prediction

strategies (HMPS) into an MOEA based on decomposition (MOEA/D) [44] to solve DMOPs. Many  
 40 existing hybrid methods utilize the response strategies at each change without justifying the suitability  
 of the strategies to the changes, which may result in less efficient reuse of the historical information.  
 To address this issue, in this paper we describe the design of an approach for distinguishing whether  
 an environmental change is similar to historical changes, aimed at improving the reuse efficiency of  
 the stored information. If a change is not similar to any historical changes, a differential prediction  
 45 method [27, 40, 30, 48] is initialized to handle the change; otherwise, a memory-based technique is  
 applied to handle the change. We devise a novel memory-driven prediction strategy to respond to  
 similar changes and speed up the convergence. The tactic of utilizing the advantages of both strategies  
 is intended to improve the capability of the algorithm to track the moving PF/PS while maintaining a  
 good population distribution. The resultant algorithm, namely, MOEA/D-HMPS, was evaluated and  
 50 compared with state-of-the-art DMOEAs using 14 benchmark problems. The experimental results  
 demonstrate that MOEA/D-HMPS is superior or comparable to the DMOEAs with which it was  
 compared. The main contributions of this study are summarized as follows.

1. A fixed detector approach was utilized to detect changes efficiently. A new method was designed  
 that compares the objective values of fixed detectors at different time instances to determine  
 55 whether a detected change is similar to historical changes, based on which the reuse efficiency of  
 the stored historical information is improved.
2. Two response mechanisms were developed to deal with similar and dissimilar changes. In our  
 method, for dissimilar changes, a differential prediction mechanism based on the previous evo-  
 lution direction of the population is used to predict the new location of optimal solutions. To  
 60 handle similar changes, a second mechanism driven by memory (the stored information on opti-  
 mal solutions) is applied to relocate the optimal solutions in new environments.
3. To ensure the robustness of HMPS, in our method half of the population individuals with the best  
 ranking values are predicted via a differential prediction or memory-driven prediction strategy.  
 For the other half, a roulette strategy is adopted to decide whether an existing individual is  
 65 reused or replaced by a randomly generated individual, which alleviates the impact of prediction  
 errors, especially in the case of sharp or irregular changes.
4. HMPS was combined with MOEA/D to utilize the advantages of both strategies to handle  
 DMOPs.

The rest of this paper is organized as follows. Section 2 provides a brief review of related work.  
 70 Section 3 describes the proposed algorithm, MOEA/D-HMPS. Section 4 introduces the test bench-  
 mark problems, comparative algorithms, performance metrics, and parameter settings used in the  
 experiments. Section 5 presents the experimental results. Finally, Section 6 concludes this study and  
 introduces potential future directions.

## 2. Related Work

75 Recently, a number of DMOEAs have been proposed to handle DMOPs [46, 26, 13, 20, 18]. In general, the existing DMOEAs are composed of three key components: change detection, change response, and an underlying MOEA framework. Change detection is designed to detect changes. When a change has been detected, the change response component can employ specific dynamic handling techniques to adapt the search behavior quickly in the new environment. In between every two changes,  
80 a DMOP is essentially a stationary MOP, where a promising MOEA framework is needed to solve the problem. The representative studies on designing the three components of DMOEAs are reviewed in the following sections.

### 2.1. Change Detection

There are two frequently used means of detecting changes in DMOEAs: 1) the re-evaluation of  
85 some dedicated detectors [46, 10, 20, 16], and 2) the identification of algorithm behavior via examining population statistical information [11, 28]. The first approach re-evaluates the detectors at every evolution generation to monitor the change in their objective values. This type of detection method is easy to implement but requires additional function evaluations and is not applicable in a noisy environment. The dedicated detectors can comprise fixed [26, 13] or unfixed detectors [46, 20, 40].  
90 Fixed detectors are randomly generated in the decision space and remain the same at every generation, whereas unfixed detectors are randomly selected from the existing population and change at every generation. The second approach checks the discrepancy between the population statistical information and the algorithms' inherited behavior. This method requires no function evaluations but may cause false positives, leading to an overreaction when no change occurs [28].

### 2.2. Change Response

When a change is detected, promising dynamic handling techniques should be adopted to respond to the change. Such techniques include mainly prediction-based [17, 47, 26], memory-based [21, 31], diversity-introduction [10, 1, 2], and multiple population methods [33, 14, 25]. Depending on the behavior of the algorithms, these techniques can be divided into two categories: techniques for accelerating  
100 convergence, and techniques for improving diversity.

#### 2.2.1. Techniques for accelerating convergence

Prediction-based and memory-based methods are effective strategies for solving DMOPs with predictable properties and cyclic changes, respectively. Both types of method are designed to promote the convergence rate.

##### 105 a) Prediction-based methods

If the dynamic changes of DMOPs are predictable, prediction-based strategies can learn the change patterns from the past changes to predict the new locations of the optimal solutions [46, 17, 21]. Thus, the optimal solutions can quickly converge to the new PS. For example, Hatzakis and Wallace [17]

proposed a feed-forward prediction strategy (FPS) to store the anchor points (boundary points) of the  
110 obtained PF. On the basis of the time series of the anchor points, FPS uses an autoregressive (AR)  
model to predict the new location of each anchor point and track the new PF. Zhou et al. [46] presented  
a population prediction strategy (PPS) to divide a PS into a centroid point and a manifold. Based on  
a series of stored center points, PPS adopts a univariate AR model to predict the subsequent center,  
and the subsequent manifold is estimated by using the previous manifolds. The entire population can  
115 then be reinitialized by the predicted center and the estimated manifold. Wu et al. [40] introduced  
a directed search strategy (DSS) in which the moving direction of the PS centers at the previous two  
time steps are used to predict the new locations of optimal solutions. Muruganantham et al. [26]  
advanced a prediction-based strategy in which a linear discrete time Kalman filter (KF) was adopted  
to estimate the locations of the Pareto-optimal solutions in the new environment. The resultant  
120 algorithm, MOEA/D-KF, shows a good ability to relocate the individuals closer to the new PS and  
achieves a promising performance. Jiang and Yang [20] proposed a steady-state and generational  
evolutionary algorithm (SGEA). When a change is detected, SGEA combines the obtained historical  
optimal information with the a priori information collected from the new environment to relocate the  
optimal solutions.

#### 125 b) Memory-based methods

When the changes in DMOPs are cyclic, reusing the stored optimal solutions or other optimal in-  
formation in the new environment could achieve faster convergence [21, 27]. Memory-based approaches  
normally store the historical optimal information over the run and reuse the information subsequently  
when the new optima are sufficiently close to the historical ones. Memory-based strategies can be  
130 implemented explicitly or implicitly [4]. In explicit memory methods, the stored optimal individuals  
are reintroduced into the reinitialized population when a change occurs, whereas in implicit memory  
methods redundant representations are employed to generate individuals in the new environment [4].  
For instance, Wang et al. [37] proposed two memory schemes in which previous optimal solutions are  
stored in an archive. When a change occurs, the first scheme randomly selects individuals from the  
135 archive and introduces them into the population. The second scheme also randomly selects individuals  
from the archive, but the selected individuals are modified using Gaussian local search before they  
are injected into the population. Koo et al. [21] presented a new selective memory technique to store  
the geometric centroid and centroid variance of the outdated archive. When a change has been de-  
tected, a memory item with a small crowding distance is selected to help generate individuals in the  
140 new environment. Peng et al. [27] presented novel prediction and memory strategies (PMS) that can  
effectively reuse the stored individuals. PMS store the non-dominated solutions into a memory pool at  
every time step. In the face of a change, the stored individuals are reevaluated, and the non-dominated  
individuals are added to the population.

The three methods discussed above are exploitations of memory-based techniques for solving  
145 DMOPs and they have been shown to be very competitive for handling DMOPs. However, it is  
noteworthy that more effort is still needed to improve the reuse efficiency of the historical optimal

information.

In addition to the prediction and memory techniques mentioned above, other methods exist to improve the convergence rate. For instance, Sen et al. [13] used an inverse modeling approach to generate individuals in the decision space, which helps guide the search toward a promising area. Jiang et al. [18] proposed a novel algorithm framework integrating transfer learning and traditional MOEAs (Tr-DMOEA) to solve DMOPs. Tr-DMOEA re-initializes an effective population by utilizing techniques using transfer learning on past experience to promote the evolutionary process.

### 2.2.2. Techniques for improving diversity

Good diversity maintenance is an additional pivotal issue for handling DMOPs. The changes in DMOPs may cause severe diversity loss. Therefore, it is essential to introduce or maintain diversity in the dynamic response. For example, Deb and Karthik [10] adopted a dynamic version of the non-dominated sorting genetic algorithm II (DNSGA-II) to handle DMOPs. To increase diversity after a change occurs, DNSGA-II replaces a part of the population individuals with either mutated or randomly generated solutions. Goh and Tan [14] proposed a new dynamic coevolutionary algorithm (dCOEA) in which competitive and cooperative mechanisms are combined. dCOEA uses a competitive mechanism to introduce randomly generated individuals into the competition pool to increase diversity. Ruan et al. [30] devised an effective diversity maintenance strategy (DMS) to enhance the diversity of an MOEA for solving DMOPs. According to the moving direction of the decision vectors of the previous maximum and minimum values in each dimension, DMS estimates the boundary of each dimension of the decision variable in the new environment and then randomly generates individuals in the estimated regions. These diversity-based strategies can effectively compensate the diversity loss due to the dynamic changes in DMOPs. However, the introduction of excessive diversity may negatively affect the convergence performance and result in evolutionary stagnation [3].

### 2.3. Stationary Multiobjective Optimization Problem Optimization

In between every two changes, a DMOP is naturally a static MOP. Therefore, a robust MOEA framework is also critical for addressing the underlying MOP. Conventional MOEAs are usually applied directly or are slightly modified to solve DMOPs, such as the non-dominated sorting genetic algorithm II (NSGA-II) [9], MOEA/D [44], and the regularity model-based multiobjective estimation of distribution algorithm (RM-MEDA) [45]. Because dynamic changes may slow down the convergence rate or cause diversity loss, the backbone MOEA should be able to achieve fast convergence and maintain good distribution when solving a static MOP.

From the above discussion, it follows that the different components in a DMOEA make their own contributions in different aspects of solving DMOPs. In this study, an attempt was made to hybridize memory and prediction strategies within the framework of MOEA/D, with the intention of taking advantage of each component to improve the solution quality of DMOPs. The proposed algorithm, referred to as MOEA/D-HMPS, is described in the following section.

---

**Algorithm 1** Overall Framework of MOEA/D-HMPS

---

```
1: Input:  $N$  (Population size),  $t$  (time step)
2: Output: a series of approximated PFs and PSs;
3: Set  $t = 0$ ,  $D = \emptyset$ ,  $C = \emptyset$ ,  $flag = false$  and  $index = -1$ ;
4: Initialize a population  $P_t = \{x_1, \dots, x_N\}$ ;
5: while stopping criterion is not met do
6:   if change detected then
7:     Compute  $\bar{F}_{t+1}$ ;
8:      $(flag, index) = isSimilar(D, \bar{F}_{t+1})$ ;
9:     Update  $D$  and  $C$  using the memory strategy;
10:     $P_{t+1} = ChangeReaction(flag, index, C, P_t)$ ;
11:     $t = t + 1$ ;
12:   else
13:     Optimize the  $t$ -th MOP using MOEA/D-DE[23];
14:   end if
15: end while
```

---

### 3. Proposed Multiobjective Evolutionary Algorithm based on Decomposition with a Hybrid of Memory and Prediction Strategies (MOEA/D-HMPS)

#### 185 3.1. Overall Framework of Multiobjective Evolutionary Algorithm based on Decomposition with a Hybrid of Memory and Prediction Strategies (MOEA/D-HMPS)

The overall framework of MOEA/D-HMPS is presented in Algorithm 1. MOEA/D-HMPS starts with an initial population  $P$ , an archive  $D$  for storing the mean values in each objective dimension of all detectors at historical time steps, and an archive  $C$  to store the population centers at historical time steps.  $D$  and  $C$  are initially empty sets and are updated by the proposed memory strategy during the evolutionary search. In Algorithm 1,  $flag$  is a Boolean variable indicating whether the current change is similar to historical changes,  $index$  represents the position index of the similar environment in  $D$  and  $C$ , and  $\bar{F}_{t+1}$  records the mean values of all detectors at time step  $t+1$ . If a change is detected in a DMOP, MOEA/D-HMPS can determine whether the current environmental change is similar to the historical changes by comparing  $\bar{F}_{t+1}$  with the elements in archive  $D$ , and  $flag$  and  $index$  are updated with the corresponding values (line 8). Then, the change response strategy handles similar or dissimilar changes according to the values of  $flag$  and  $index$  (line 10). If no change occurs, the underlying MOEA optimizes the  $t$ -th stationary MOP (line 13). In the following subsections, the key components of MOEA/D-HMPS, including change detection, change identification, memory strategy, change reaction, and the underlying MOEA optimizer, are described in detail.

### 3.2. Change Detection

An advanced change detection strategy should be able to estimate the degree of a change and provide useful information for change identification. To detect environmental changes and identify similar changes effectively, a fixed detector method [13] is adopted in MOEA/D-HMPS, where the detectors are randomly generated in the decision space and remain the same at every evolution generation. Suppose that noise is ignored in the objective function evaluation; then, the objective values of the fixed detectors are reevaluated in every generation and compared with the previously stored values. If a mismatch exists between the previous and the recalculated objective values of the detectors, then a change is detected.

### 3.3. Change Identification

This subsection introduces a novel method to distinguish similar from dissimilar changes, which can help improve the dynamic reaction effectively. In general, if the tracked PS and PF remain the same at two different time steps, the two dynamic environments are considered similar, meaning the optimal information obtained can be reused in subsequent similar environments. Thus, the convergence performance can be promoted. Because MOEA/D-HMPS uses a fixed detector approach, meaning the member detectors are the same at every generation, the objective values of the detectors at two different time steps can be compared to determine whether the two environments are similar. To implement this, the objective values of the detectors in historical changes must be stored. In the **Supplementary Materials (Section 1)**, an experimental study is described, which was conducted to evaluate the influence of storing different historical information. Based on the experimental results, in the archiving strategy, we chose to store the mean value of each objective dimension for all detectors at time step  $t$ , denoted by  $\bar{F}_t = (\bar{f}_1, \bar{f}_2, \dots, \bar{f}_m)^T$ , and the center point of a population at time step  $t$ , denoted by  $C_t$ . A series of  $\bar{F}_t$  forms an archive set  $D$ , and a series of  $C_t$  composes  $C$ .

$\bar{F}_t$  is calculated at different time steps and saved into archive  $D$  (The storing procedure of  $\bar{F}_t$  is described in Subsection 3.4.2.) When a new change is detected,  $\bar{F}_t$  of the current environment, denoted by  $\bar{F}_c$ , is compared with the previous  $\bar{F}_t$  stored in set  $D$  to determine whether a similar change exists. The detailed procedure is shown in Algorithm 2, where  $m$  represents the number of objectives and  $\epsilon$  is a pre-defined threshold. If the differences of all objectives are less than  $\epsilon$ , the  $i$ -th stored environment and the current environment are deemed similar. Accordingly, *flag* and *index* are set to *true* and  $i$ , respectively (lines 5 to 12 of Algorithm 2).

### 3.4. Memory Strategy

Memory-based approaches are effective when solving DMOPs with cyclic changes. Many DMOEAs adopt memory-based approaches to handle DMOPs [21, 31, 27, 4], where non-dominated solutions or optimal information extracted from the current population are stored in a memory pool and subsequently reused in new environments. However, such memory-based approaches suffer two shortcomings. 1) The non-dominated solutions stored in the memory may become outdated after a change occurs,



---

**Algorithm 2**  $isSimilar(D, \bar{F}_c)$ 

---

```
1: Input:  $D$  (a set of  $\bar{F}_t$  at historical time steps) and  $\bar{F}_c$  ( $\bar{F}_t$  at current change)
2: Output:  $flag$  (a Boolean value) and  $index$  (similar position index)
3: for  $i = 1$  to  $|D|$  do
4:   Set  $count = 0$ ;
5:   for  $j = 1$  to  $m$  do
6:     if  $|D_i^j - \bar{F}_c^j| < \epsilon$  then
7:        $count++$ ;
8:     end if
9:   end for
10:  if  $count == m$  then
11:    Set  $flag = true$  and  $index = i$ ;
12:    Break;
13:  end if
14: end for
15: Return  $flag$  and  $index$ ;
```

---

meaning they may not provide sufficient and effective historical optimal information and waste the space resource. 2) The stored historical optimal individuals or information are not efficiently reused in the new environment. To overcome the aforementioned shortcomings, in this study an approach was devised to identify similar changes, as described in Section 3.3, based on which the stored optimal information can be reused more correctly. Additionally, the center point of the obtained optimal population is stored to represent the historical optimal information.

#### 3.4.1. Stored historical information

From the above description, it is easy to understand that the proposed memory strategy stores: 1)  $\bar{F}_t$  at different time steps, which is used to identify similar environments; and 2) the population centers at different time steps. If a new change is a similar change, the stored population centers are reused to guide the generation of individuals in the new environment. The proposed memory strategy requires  $D$ , i.e., a series of  $\bar{F}_t$ , and  $C$ , i.e., a series of  $C_t$ .

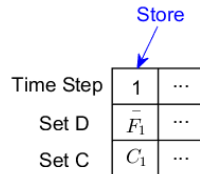




Figure 1: Storage of  $\bar{F}_1$  and  $C_1$  at the end of the first time step on the occurrence of the first change.



Time Step	1	2	...	k	...
Set D	$\bar{F}_1$	$\bar{F}_2$	...	$\bar{F}_k$	...
Set C	$C_1$	$C_2$	...	$C_k$	...

Figure 2: Addition of  $\bar{F}_k$  and  $C_k$  at the end of the  $k$ -th time step if no past change exists that is similar to the  $k$ -th change.



Time Step	1	2	3	...
Set D	$\bar{F}_1$	$\bar{F}_2$	$\bar{F}_3$	...
Set C	$C_1$	$C_2$	$C_3$	...

Time Step	1	k	3	...
Set D	$\bar{F}_1$	$\bar{F}_k$	$\bar{F}_3$	...
Set C	$C_1$	$C_k$	$C_3$	...

Figure 3: Replacement of  $\bar{F}_2$  and  $C_2$  in set  $D$  and set  $C$  with  $\bar{F}_k$  and  $C_k$ , respectively, at the end of the  $k$ -th time step.

### 3.4.2. Process of archiving and updating

250 This subsection describes the detailed procedure of archiving and updating in the proposed memory strategy. When the first change is detected, the two archive sets  $D$  and  $C$  are empty. Therefore,  $\bar{F}_1$  and  $C_1$  are added to  $D$  and  $C$ , respectively, as shown in Figure 1. On the detection of the  $k$ -th change,  $\bar{F}_k$  is compared with  $\bar{F}_t$  in  $D$  using Algorithm 2. If the  $k$ -th change is dissimilar to any historical changes,  $\bar{F}_k$  and  $C_k$  are copied to  $D$  and  $C$ , respectively, as shown in Figure 2; otherwise, the  $k$ -th change is similar to a historical change. As shown in Figure 3, suppose that the  $k$ -th change is similar to the second change; then, the archived  $\bar{F}_2$  and  $C_2$  in  $D$  and  $C$  are replaced by  $\bar{F}_k$  and  $C_k$ , respectively. The proposed method shows two advantages. First, parsimonious storage ensures the efficiency of the archiving, because there are no similar changes in the archive sets. Second, because the most recent similar change is recorded, the stored information is of high quality, which improves the performance of the change reaction if an additional similar change subsequently occurs. When the size of the two archive sets exceeds the predefined size, a first-in-first-out strategy is applied to update the archive set; i.e., no extra space resources are consumed.

## 3.5. Change Response

265 When a change is detected, DMOEAs should adopt efficient dynamic handling techniques to speed up the convergence and maintain good diversity. A good dynamic response strategy should be able to reinitialize a well-distributed population close to the PS of the new environment. In MOEA/D-HMPS, a new change is compared with the historical changes to observe whether it is a similar or dissimilar one, based on which two change response mechanisms are applied correspondingly.

### 3.5.1. Response to dissimilar change

270 For dissimilar changes, it is unreasonable to adopt memory-based strategies directly, because the historical information cannot guide the search toward the new PS. Instead, a prediction-based approach should be considered to relocate individuals in the promising areas that are approximate to the

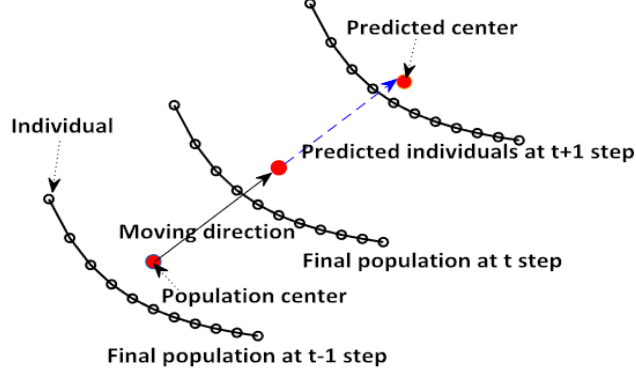


Figure 4: Prediction method in response to a dissimilar change.

new PS. In MOEA/D-HMPS, an improved differential prediction method is used to exploit historical information to predict the new location of the existing optimal solutions. Conventional differential prediction methods [27, 40, 30, 48] predict the new location of the optimal solutions based on the previous two consecutive population centers. They may not be effective for sharp and/or irregular changes and could lead to prediction errors. To solve this problem, MOEA/D-HMPS introduces a portion of the existing solutions and randomly generates solutions into the new population in response to the change, which could mitigate the effect of prediction errors. Note that there is no stored historical information when the first change is detected, meaning no differential prediction can be performed. In this case, 50% of the new population is formed by existing individuals with the best ranking in the current population and 50% by randomly generated individuals.

The differential prediction used in this study is defined as

$$x_{t+1}^k = x_t^k + C_t^k - C_{t-1}^k + \text{Gaussian}(0, d) \quad (2)$$

where  $k=1,2,\dots,n$  is the index of the decision variable and  $n$  represents the dimension of the decision vector.  $C_t^k$  and  $C_{t-1}^k$  are the  $k$ -th decision variables of the population center points at the  $t$ -th and  $(t-1)$ -th time steps, respectively.  $x_t^k$  and  $x_{t+1}^k$  represent the  $k$ -th decision variables of an individual at the  $t$ -th time step and the predicted individual at the  $(t+1)$ -th time step, respectively.  $\text{Gaussian}(0, d)$  denotes Gaussian perturbation with a mean value 0 and standard deviation  $d$ . The standard deviation  $d$  is defined as

$$d = \frac{\|C_t - C_{t-1}\|}{n} \quad (3)$$

where  $\|C_t - C_{t-1}\|$  is the Euclidean distance between centers  $C_t$  and  $C_{t-1}$ .

The prediction mechanism is illustrated in Figure 4, where a black circle indicates an individual and a series of black circles along the curve represents the distribution of individuals in the population. The red dot denotes the center point of the population. The solid arrow shows the moving direction of the population center at the previous two continuous time steps, which refers to the vector difference of the two center points. The blue dotted arrow represents the predicted evolutionary direction. Here, we employ the moving direction as the evolution direction in the new environment. The obtained solutions

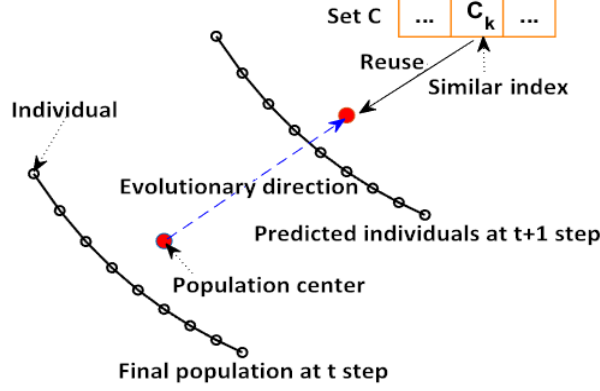


Figure 5: Prediction method adopted for responding to a change similar to a historical change.

at the  $t$ -th time step are combined with the predicted evolution direction to relocate optimal individuals at the  $(t+1)$ -th time step. Because the prediction method does not necessarily produce an accurate prediction, Gaussian perturbation is added to improve the search exploration of the population.

### 3.5.2. Response to similar change

If a similar change is detected, it is appropriate to adopt memory-based techniques to reuse the information of the previous optimal solutions in the new environment. Because the mean values in each objective dimension of the detectors in historical changes have been recorded in archive  $D$  and the population centers in historical changes have been saved in archive  $C$ , we can efficiently reuse them to respond to the similar changes.

First, a search is conducted in Algorithm 2 to identify the historical change in  $D$  that is similar to the current change. The similar change in  $D$  is indicated by *index*. Second, the population center in the similar historical environment, denoted by  $C_{index}$ , is reused given that the tracked PSs of two similar environments are similar. Finally, the existing individuals are relocated by differential prediction based on  $C_{index}$  and the current population center  $C_t$ . Unlike in the frequently used differential prediction, the predicted evolution direction here refers to the vector difference between the two center points  $C_t$  and  $C_{index}$ , rather than the center points at the previous two consecutive time steps  $C_{t-1}$  and  $C_t$ . The proposed differential prediction in this scenario is

$$x_{t+1}^k = x_t^k + C_{index}^k - C_t^k \quad (4)$$

Figure 5 demonstrates the prediction method adopted in response to a similar change. The stored  $C_{index}$  is reused to guide the reinitialization of individuals in the new environment. As in the response to dissimilar changes, a small part of the existing solutions and randomly generated solutions is also introduced into the new population to alleviate the effect of the prediction errors.

### 3.5.3. Overall procedure of change response

The overall dynamic response procedure of MOEA/D-HMPS is summarized in Algorithm 3. First, the population center at the  $t$ -th time step is calculated (line 3). Second, the individuals in the

---

**Algorithm 3** *ChangeReaction(flag, index, C, P<sub>t</sub>)*

---

```
1: Input: flag (a Boolean value), index (similar position index), C (a set of population centers), and  
   Pt (population at t-th time step)  
2: Output: Pt+1 (the reinitialized population)  
3: Calculate the center point Ct of Pt.  
4: Rank the individuals in Pt based on their fitness values.  
5: for i = 1 to N do  
6:   if xt(i) is ranked lower than 50% of the individuals in Pt then  
7:     if flag == true then  
8:       Reinitialize xt(i) using Equation (4);  
9:     else  
10:      Reinitialize xt(i) using Equation (2);  
11:    end if  
12:  else  
13:    if rand[0, 1] < 0.5 then  
14:      xt(i) is randomly generated in the decision space;  
15:    else  
16:      xt(i) is reused;  
17:    end if  
18:  end if  
19: end for  
20: Ct-1 = Ct;  
21: Return Pt+1;
```

---

population  $P_t$  of size  $N$  are sorted based on their fitness values, and the ranking value is assigned to the corresponding individual (line 4), where a lower rank indicates a better fitness value. MOEA/D-DE [23] is utilized as the underlying MOEA optimizer, and the fitness values of the individuals refer to the Tchebyshev value [35]. In the change response, half of the population individuals with the  
325 best ranking values are predicted via a differential prediction or memory-driven prediction strategy, which relocates the optimal solutions close to the new PS and maintains good distribution in the objective space (lines 6 to 11). The other half of the population is formed by the existing individuals and randomly generated individuals. A roulette strategy is adopted to decide whether the existing individual is reused or replaced by a randomly generated individual. The roulette probability is set  
330 to 0.5 (lines 13 to 17). Finally,  $C_{t-1}$  is replaced with  $C_t$ , and the reinitialized  $P_{t+1}$  is returned as the output (lines 20 and 21).

### 3.6. Underlying Multiobjective Evolutionary Algorithm

The proposed MOEA/D-HMPS employs MOEA/D-DE [23] as the underlying MOEA optimizer. MOEA/D-DE performs very well for handling continuous MOPs, being able to achieve fast convergence  
335 and a diverse spread. It has gained considerable attention for solving DMOPs. MOEA/D decomposes a problem into multiple subproblems and optimizes them simultaneously based on the neighboring relations. The definition of neighboring relations is based on the distances of the individuals' weight vectors in the objective space. The fitness value of an individual is the aggregational function value. Here, a classic Tchebycheff approach is adopted as the aggregation method because of its simplicity  
340 and good optimization performance.

## 4. Experimental Design

### 4.1. Test Problems and Compared Algorithms

In this section, the assessment of the proposed MOEA/D-HMPS using 14 frequently used benchmark DMOPs involving the FDA [11], dMOP [14], and F [46] test suites is described.

1) The FDA and dMOP test problems have linear linkages between the decision variables. They  
345 include three relatively difficult problems: FDA3, FDA5, and dMOP3. In FDA3 and FDA5, the density distribution of the solutions along the PF is time varying. It is difficult for DMOEAs to maintain a good distribution on these two test problems. In dMOP3, the diversity-related variable is randomly selected over time, which could cause a dramatic decrease in the population diversity. Note that a  
350 modified FDA2 defined in MOEA/D-KF [26] was used in this study instead of the original version.

2) The test suite F consists of 10 problems. F1–F4 are partly adopted from the FDA and dMOP test suites. F5–F10 were recently proposed to provide problems having nonlinear linkages between the decision variables. Unlike those in the FDA and dMOP test problems, the dynamic changes in the F test problems are sharp and irregular. In particular, in F9 jumping changes occasionally occur, and in  
355 F10 the PFs of two consecutive time steps are completely different. These difficulties may lead to some errors in the response mechanism, resulting in a poor convergence performance of the algorithms.

The proposed MOEA/D-HMPS is compared with three representative state-of-the-art DMOEAs: the population prediction strategy (PPS) [46], MOEA/D-DE with Kalman Filter prediction (MOEA/D-KF) [26], and a steady-state and generational evolutionary algorithm (SGEA) [20]. PPS is frequently  
360 used as compared with other DMOEAs. MOEA/D-KF and SGEA are two newly proposed strategies, which have shown promising performances for solving DMOPs. The comparative algorithms were previously described in Subsection 2.2.1.

### 4.2. Performance Metrics

A number of performance metrics have been used to evaluate DMOEAs [34, 6, 15, 32, 41]. How-  
365 ever, most of these were originally designed to assess static MOEAs, such as maximum spread (MS) [15], Schott's spacing metric (SP) [32], and inverted generational distance (IGD) [41]. To assess the

performance of DMOEAs effectively, a modified IGD (MIGD) metric [46] and a modified hypervolume (MHV) [5] were adopted in this experimental study. Both metrics can measure well the performance of the algorithms in terms of convergence and diversity over time.

#### 370 4.2.1. Modified inverted generational distance

Before MIGD is introduced, the basic IGD is described for the sake of the reader's understanding of MIGD. IGD is a widely used comprehensive indicator of a MOEA's performance, which evaluates both the convergence and diversity of the obtained solutions. Let  $PF_t$  be a set of uniformly distributed points in the true PF and  $PF_t^*$  be an approximation of the true PF obtained by the algorithm at the  
375  $t$ -th time step. The IGD metric is defined as

$$IGD(PF_t, PF_t^*) = \frac{\sum_{v \in PF_t} d(v, PF_t^*)}{|PF_t|} \quad (5)$$

where  $d(v, PF_t^*) = \min_{u \in PF_t^*} \|F(v) - F(u)\|$  is the minimum Euclidean distance between a point  $v$  in  $PF_t$  and points in  $PF_t^*$ . A lower IGD value suggests better convergence and distribution of the obtained solutions.

The MIGD [46] metric is defined as the average of the IGD values over all time steps in a run of a  
380 DMOEA; i.e.,

$$MIGD = \frac{1}{|T|} \sum_{t \in T} IGD(PF_t, PF_t^*) \quad (6)$$

where  $T$  is a set of discrete time steps in a run and  $|T|$  calculates the cardinality of  $T$ .

#### 4.2.2. Modified hypervolume

The MHV measure is a modification of the static measure HV [47], which computes the hypervolume of the area dominated by the obtained  $PF^*$ . Similarly, the MHV metric is defined as the average of  
385 the HV values over all time steps in a run of a DMOEA; i.e.,

$$MHV = \frac{1}{|T|} \sum_{t \in T} HV(PF_t^*) \quad (7)$$

where  $T$  is a set of discrete time steps in a run and  $|T|$  is the cardinality of  $T$ . The reference point for the computation of hypervolume is  $(z_1 + 0.1, z_2 + 0.1, \dots, z_m + 0.1)$ , where  $z_j$  is the maximum value of the  $j$ -th objective of the true PF at time  $t$  and  $m$  is the number of the objectives. A larger MHV value indicates better convergence and distribution of the obtained solutions.

#### 390 4.3. Parameter Settings

The parameters of the compared algorithms were configured following the original references. The key parameters of these algorithms are highlighted as follows.

1) Environmental change: Changes are assumed to occur in between generations, and the conversion of the generation number into time is computed as [11]

$$t = \frac{1}{n_t} \lfloor \frac{\tau}{\tau_t} \rfloor \quad (8)$$

395 where  $\tau$ ,  $n_t$ , and  $\tau_t$  represent the generation counter, the severity of change, and the frequency of change, respectively. The change frequency  $\tau_t$  and severity  $n_t$  determine the difficulty of a problem. Following PPS and MOEA/D-KF [46, 26], these two parameters were reasonably set to  $\tau_t=30$  and  $n_t=10$ , respectively.

2) Dimension of decision variables: The dimension of decision variables was set to 20 for all test problems. 400

3) Population size: The population size was set to 100 for biobjective problems and 105 for triobjective problems.

4) Change detection: In SGEA and PPS, 10 individuals are randomly selected from the existing population as the detectors. In comparison, both MOEA/D-HMPS and MOEA/D-KF adopt the fixed 405 detector approach, where 10 randomly generated individuals in the decision space are employed to detect the changes.

5) Stopping criterion and the number of runs: The algorithms stop after the predefined number of iterations is reached. In the experiment, each algorithm was run 20 times independently on each test problem. The total number of generations was set to  $120 * \tau_t$ , which ensured 120 environmental 410 changes in each run.

6) Parameter setting of MOEA/D-HMPS: In addition to the aforementioned parameters, the threshold  $\epsilon$  introduced in Algorithm 2 was empirically set to  $10^{-4}$ . In general, if two environments are similar, the tracked PS and PF in these two environments are the same and the objective values of the detectors in these two environments are approximately equal to each other. Considering the noise in individual 415 reevaluations [46, 20], the setting of  $\epsilon$  is reasonable. The maximum memory size of the two archive sets  $D$  and  $C$  was set to 50.

## 5. Experimental Results and Discussions

This section presents the experimental results of the compared algorithms on the benchmark problems. The statistic results of MIGD and MHV values for all test instances are presented in Tables 1 420 and 2, respectively. The best result of the four algorithms on each test problem is in bold face. The significance difference between the performance of MOEA/D-HMPS and that of the other algorithms was tested using a Wilcoxon rank-sum test [39] at a significance level of 0.05. To visualize the performance of the algorithms, the average MIGD values on some representative problems at every time step in the 20 independent runs are plotted in Figures 6 and 7. A nested graph is also provided in 425 each subfigure by zooming in the original graph to show the details, which helps to distinguish the differences of the algorithms' performance.

### 5.1. Results on FDA and dMOP Problems

Both FDA and dMOP test instances have linear correlation among decision variables. In the MIGD and MHV values reported in Tables 1 and 2, it is observed that MOEA/D-HMPS shows a better performance than the other three algorithms in most test problems. The results in Figure 6 suggest that 430



Table 1: Mean and standard deviation values of modified inverted generational distance metric obtained by four dynamic multiobjective evolutionary algorithms on all test instances

Problem	PPS	SGEA	MOEA/D-KF	MOEA/D-HMPS
FDA1	1.179E-2(4.916E-3)(-)	1.077E-2(1.034E-3)(-)	7.572E-3(2.703E-4)(-)	<b>5.621E-3(2.062E-4)</b>
FDA2	1.055E-2(1.270E-3)(-)	1.057E-2(3.090E-3)(-)	5.867E-3(2.886E-4)(-)	<b>5.423E-3(5.875E-4)</b>
FDA3	2.554E-2(1.425E-2)( $\approx$ )	2.114E-2(2.192E-3)(-)	2.129E-2(1.152E-3)(-)	<b>1.559E-2(9.470E-4)</b>
FDA4	1.431E-1(2.659E-3)(-)	8.910E-2(1.711E-3)(-)	8.839E-2(1.236E-3)(-)	<b>8.150E-2(2.502E-3)</b>
FDA5	1.979E-1(9.184E-3)(-)	<b>1.245E-1(1.813E-3)(+)</b>	1.666E-1(1.012E-2)(-)	1.439E-1(1.297E-2)
dMOP1	3.546E-2(3.388E-2)(-)	9.797E-3(1.739E-3)(-)	8.581E-3(9.545E-4)(-)	<b>8.054E-3(1.036E-3)</b>
dMOP2	1.780E-2(1.119E-2)(-)	1.280E-2(1.743E-3)(-)	8.531E-3(5.300E-4)(-)	<b>6.374E-3(3.713E-4)</b>
dMOP3	2.309E-2(1.381E-3)(+)	9.243E-2(5.075E-3)(-)	<b>2.243E-2(1.945E-3)(+)</b>	2.320E-2(2.058E-3)
F5	3.664E-2(1.757E-2)(-)	6.281E-1(1.909E-2)(-)	2.971E-2(4.139E-3)(-)	<b>2.122E-2(5.577E-3)</b>
F6	6.632E-2(2.287E-2)(-)	3.119E-1(1.736E-2)(-)	6.824E-2(1.543E-2)(-)	<b>5.324E-2(9.546E-3)</b>
F7	3.619E-2(9.279E-3)(-)	3.265E-1(7.230E-3)(-)	3.700E-2(5.999E-3)(-)	<b>2.926E-2(6.961E-3)</b>
F8	4.093E-1(1.709E-2)(-)	2.582E-1(4.376E-3)(-)	1.078E-1(1.804E-3)(-)	<b>1.058E-1(1.935E-3)</b>
F9	1.142E-1(4.442E-2)(-)	1.257E+0(8.146E-2)(-)	1.253E-1(3.105E-2)(-)	<b>2.138E-2(2.496E-3)</b>
F10	1.132E-1(2.342E-2)(-)	4.427E-1(1.319E-2)(-)	<b>4.285E-2(7.823E-3)(+)</b>	8.137E-2(8.934E-3)

+/-/ $\approx$  indicates that the performance of the corresponding algorithm is significantly better than, significantly worse than, and comparable to that of MOEA/D-HMPS, respectively, at the 5% significance level under the Wilcoxon rank-sum test.

MOEA/D-HMPS can efficiently relocate the solution set closer to the new PS than the other algorithms, leading to better convergence. The AR prediction model adopted in PPS needs to be trained with a sufficient number of stored population center points; however, in reality, at the early stages of the evolutionary process, the quality of the center points stored is not sufficiently good to support accurate prediction, and the performance of PPS thus deteriorates. The prediction methods of SGEA, MOEA/D-KF, and MOEA/D-HMPS need no training to achieve a good performance. Nevertheless, the prediction model of SGEA, which is simply based on the moving step size and moving direction, may lead to poor prediction quality. The prediction model of MOEA/D-KF, which heavily relies on the accuracy of the Kalman filter prediction, tends to yield errors when dealing with a nonlinear dynamic system. Prediction errors negatively affect the algorithms' performance. The performance of the algorithms is further analyzed according to the different characteristics of the test problems as follows.

The PSs of FDA1, FDA4, and dMOP2 change in a sinusoidal pattern. According to the symmetry and periodicity of the sinusoidal curve, it is straightforwardly deduced that there are many similar changes in these three test instances. Similarly, FDA2 also involves many similar changes, as the changes in FDA2 are cyclic. As a result, the proposed MOEA/D-HMPS, equipped with a memory-driven prediction strategy, unsurprisingly shows a superior performance on these four problems, as demonstrated in Tables 1 and 2 and Figures 6(a), 6(b), 6(d), and 6(f).

dMOP1 is a typical Type III DMOP in that its PS remains fixed during the change process. SGEA reuses half of the existing population individuals to reinitialize the population in the new environment.

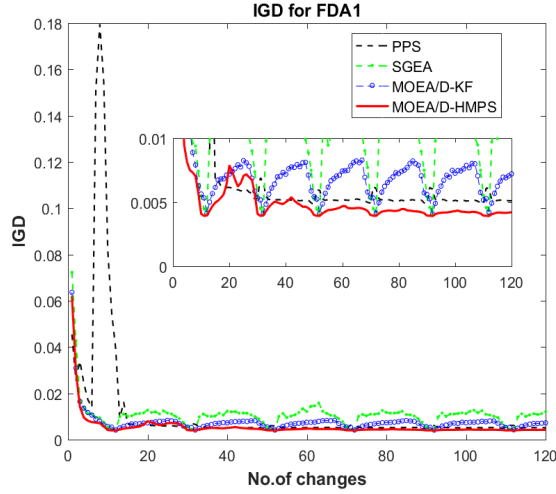
Table 2: Mean and standard deviation values of modified hypervolume metric obtained by four dynamic multiobjective evolutionary algorithms on all test instances

Problem	PPS	SGEA	MOEA/D-KF	MOEA/D-HMPS
FDA1	8.570E-1(6.600E-3)(-)	8.582E-1(7.870E-4)(-)	8.619E-1(3.790E-4)(-)	<b>8.669E-1(3.242E-4)</b>
FDA2	6.624E-1(1.759E-3)(-)	6.671E-1(2.888E-3)(-)	6.705E-1(4.535E-4)(-)	<b>6.718E-1(7.385E-4)</b>
FDA3	1.289E+0(2.689E-2)(-)	1.464E+0(2.673E-3)(-)	<b>1.509E+0(8.930E-3)(+)</b>	1.496E+0(5.604E-3)
FDA4	4.705E-1(4.003E-3)(-)	6.360E-1(3.594E-3)(-)	6.185E-1(2.779E-3)(-)	<b>6.425E-1(5.035E-3)</b>
FDA5	4.939E+0(8.363E-3)(-)	5.037E+0(2.805E-3)(-)	5.022E+0(4.744E-3)(-)	<b>5.042E+0(5.168E-3)</b>
dMOP1	6.487E-1(1.879E-2)(-)	<b>6.695E-1(1.608E-3)(+)</b>	6.672E-1(9.631E-4)(-)	6.686E-1(9.780E-3)
dMOP2	6.568E-1(7.364E-3)(-)	6.585E-1(1.703E-3)(-)	6.643E-1(6.740E-4)(-)	<b>6.696E-1(5.092E-4)</b>
dMOP3	8.336E-1(2.549E-3)( $\approx$ )	7.742E-1(6.229E-3)(-)	<b>8.370E-1(2.105E-3)(+)</b>	8.348E-1(2.147E-3)
F5	7.215E-1(1.755E-2)(-)	1.974E-1(6.012E-3)(-)	7.233E-1(5.722E-3)(-)	<b>7.411E-1(6.765E-3)</b>
F6	7.111E-1(1.797E-2)(-)	4.488E-1(1.2290E-2)(-)	6.980E-1(1.161E-2)(-)	<b>7.210E-1(7.440E-3)</b>
F7	7.334E-1(1.095E-2)( $\approx$ )	4.432E-1(6.471E-3)(-)	7.221E-1(6.591E-3)(-)	<b>7.373E-1(6.036E-3)</b>
F8	1.642E-1(1.113E-2)(-)	4.578E-1(4.786E-3)(-)	5.688E-1(2.875E-3)(-)	<b>5.752E-1(3.106E-3)</b>
F9	6.354E-1(3.604E-2)(-)	2.517E-1(1.162E-2)(-)	6.624E-1(1.635E-2)(-)	<b>7.418E-1(3.481E-3)</b>
F10	5.984E-1(2.233E-2)(-)	3.383E-1(5.298E-3)(-)	<b>7.018E-1(1.136E-2)(+)</b>	6.429E-1(1.147E-2)

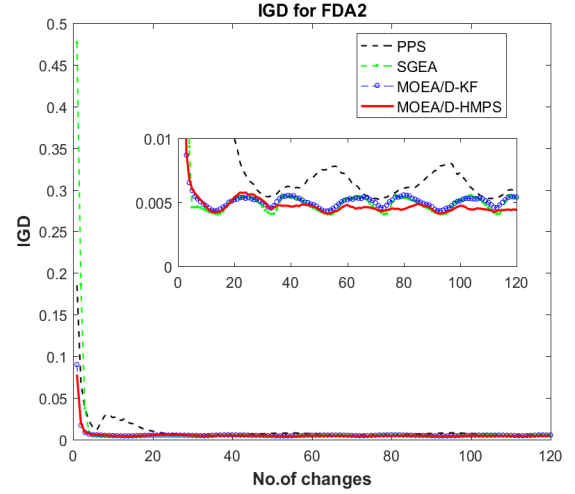
+/-/ $\approx$  indicates that the performance of the corresponding algorithm is significantly better than, significantly worse than, and comparable to that of MOEA/D-HMPS, respectively, at the 5% significance level under the Wilcoxon rank-sum test.

It is deemed suitable for solving such problems. In the results shown in Tables 1 and 2, it can be seen that the MIGD and MHV values are inconsistent on dMOP1; both show that SGEA and MOEA/D-HMPS obtain competitive performances on dMOP1. According to the MHV results, SGEA outperforms MOEA/D-HMPS. However, according to Table 1 and Figure 6(e), SGEA shows a poorer performance than MOEA/D-KF and MOEA/D-HMPS at the early stage because of the low quality of the obtained optimal solutions. Although its performance improves at a later stage, the final results of SGEA in dMOP1 remain inferior to MOEA/D-KF and MOEA/D-HMPS.

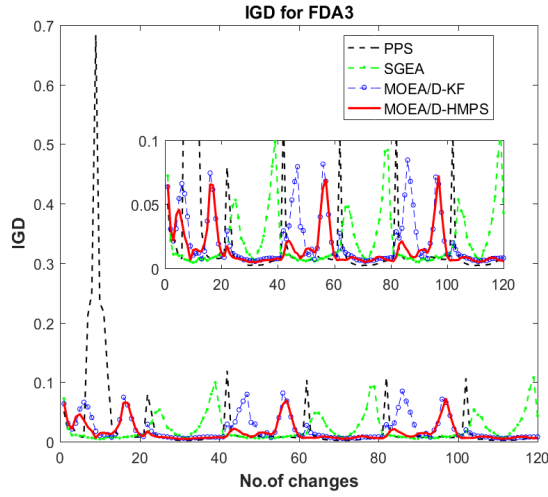
In FDA3 and FDA5, the density distribution of the solutions along the PF is time-variant, which poses a significant challenge to DMOEAs. Both FDA3 and FDA5 have cyclic changes. They differ in that FDA3 is biobjective, whereas FDA5 is triobjective. In Tables 1 and 2, it can be seen that the MIGD and MHV values of the algorithms are inconsistent in FDA3 and FDA5. The changes in the density distribution of solutions along the PF have some effect on the calculation of MIGD and MHV. On FDA3, MOEA/D-HMPS achieves faster convergence through the combined memory- and prediction-based techniques. The underlying MOEA/D is also effective for producing a uniformly distributed set of solutions. MOEA/D-HMPS is ranked in the first place and second place among the compared algorithms in Tables 1 and 2, respectively. FDA5 is a triobjective problem with a time-variant PF and PS. SGEA and MOEA/D-HMPS obtain a better performance than the other algorithms. SGEA can obtain a promising performance on FDA5 by virtue of the use of non-dominated sorting [9] and farthest-first selection [36]. On FDA5, MOEA/D-HMPS is ranked in the second place and first place among the compared algorithms in Tables 1 and 2, respectively. Overall, MOEA/D-HMPS is very



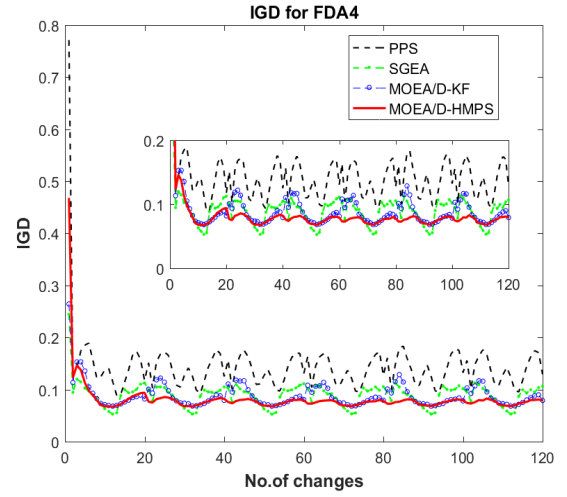
(a)



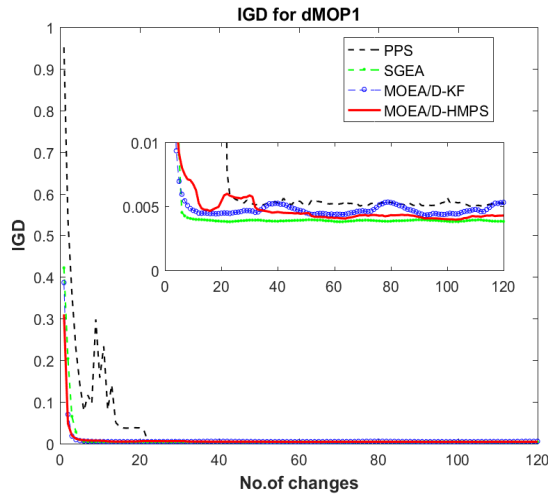
(b)



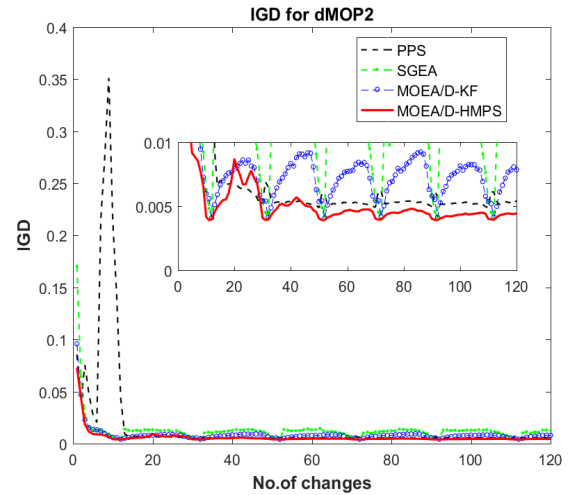
(c)



(d)



(e)



(f)

Figure 6: Average inverted generational distance values obtained by 4 algorithms over the number of changes for 20 runs.

effective when dealing with FDA3 and FDA5.

The diversity-related variables in dMOP3 change randomly over time, which may cause a great loss in diversity. SGEA does not use effective diversity introduction techniques in the change response, and therefore shows a poor performance on dMOP3. MOEA/D-KF and MOEA/D-HMPS attain better  
475 results on dMOP3, because they not only introduce random individuals in the change reaction but also adopt MOEA/D as the underlying MOEA, which can help maintain good diversity. dMOP3 has few similar changes in the changing process where PPS and MOEA/D-KF outperform MOEA/D-HMPS. It is not difficult to understand why PPS outperforms MOEA/D-HMPS on dMOP3. The underlying RM-MEDA adopted in PPS can learn the distribution of the decision variables in the stationary  
480 optimization process, which helps compensate the diversity loss effectively.

## 5.2. Results on F Problems

F test instances involve nonlinear correlation among decision variables. According to the MIGD and MHV results shown in Tables 1 and 2, MOEA/D-HMPS consistently shows a better performance than the comparative algorithms on the majority of the test problems.

485 SGEA seems to struggle to solve F test problems. The reason lies in the response mechanism of SGEA. Half of the outdated population members are reused in the new environment; however, these members are less helpful on DMOPs that have nonlinear correlation among decision variables. Moreover, the simulated binary crossover adopted in SGEA also cannot be applied to solving DMOPs with nonlinear decision variable correlations [20].

490 F5–F10 are DMOPs with cyclic changes, yielding many similar changes in an independent run. MOEA/D-HMPS is competitive in solving such problems, which explains its superiority to the other algorithms, as shown in Tables 1 and 2 and Figure 7.

On F5–F7, MOEA/D-HMPS and PPS show better performances than MOEA/D-KF and SGEA. It is clearly seen in Figures 7(a) and 7(b) that MOEA/D-HMPS dominates PPS at the early stage (in  
495 the first 20 changes), because PPS needs a long time to train the prediction model, and the obtained population centers at the early stage are of low quality. However, PPS wins at the later stage, because 1) at the later stage, sufficient high quality historical information is available to the AR model in PPS to allow accurate prediction, and 2) PPS adopts an estimation of distribution algorithm (EDA) [45] to generate offspring individuals, which is more powerful than the differential evolution (DE)  
500 operator adopted in MOEA/D-HMPS to learn the linkage among decision variables. Fortunately, the response mechanism of MOEA/D-HMPS can compensate and respond immediately to the change and reinitialize a population with higher quality than the AR model, which leads to the better overall results.

F8 is a complicated triobjective problem, the PF of which remains fixed in the changing process.  
505 F8 also has many periodic changes that fit well the response mechanism of MOEA/D-HMPS. The efficiency of MOEA/D-HMPS in handling similar changes allows it to outperform the other algorithms on F8.

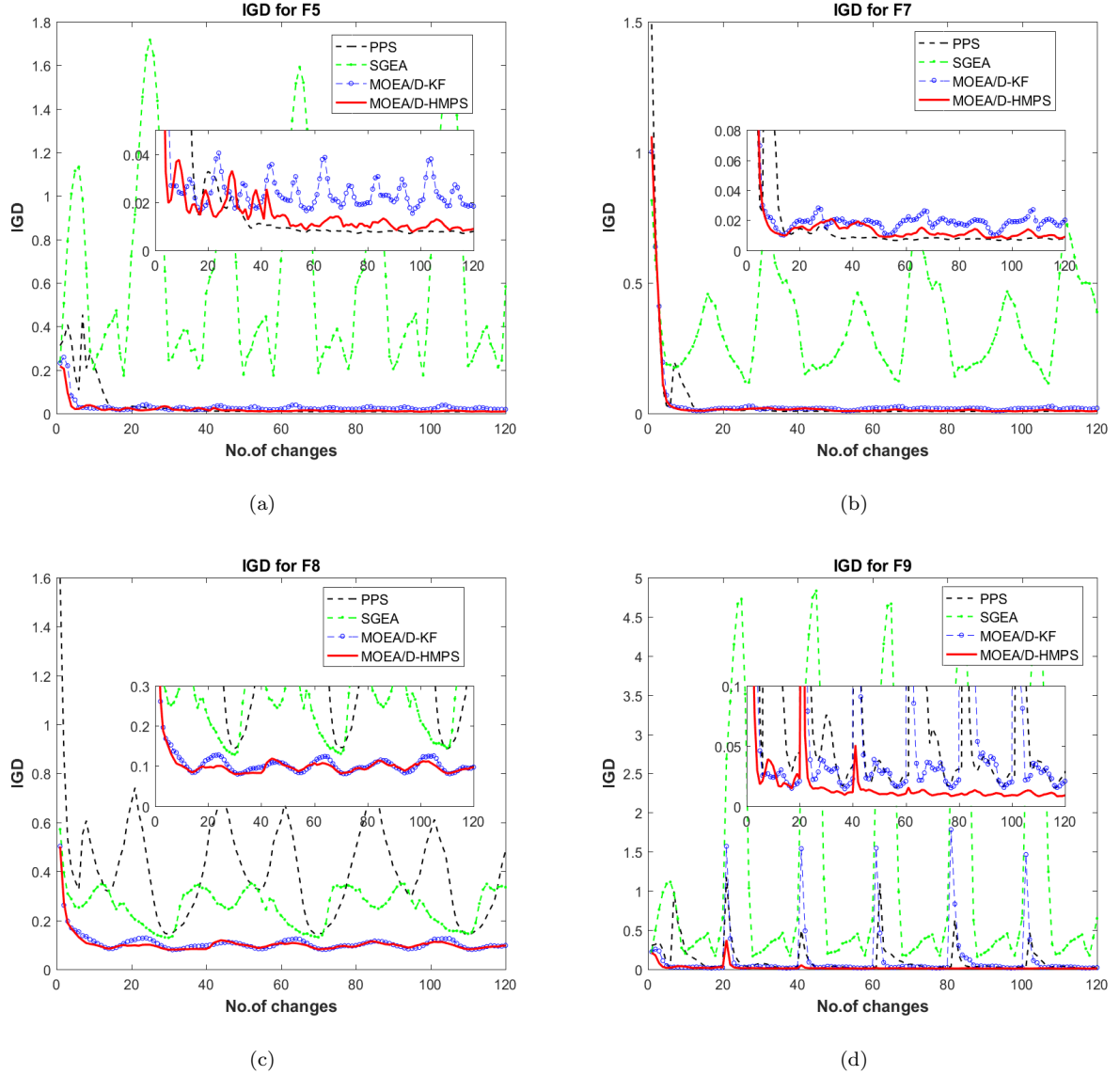


Figure 7: Average inverted generational distance values obtained by 4 algorithms over the number of changes for 20 runs.

F9 and F10 are two more complicated problems designed to challenge DMOEAs. In general, the environmental changes of the problems discussed above are smooth, whereas F9 includes occasional jumping changes of the PS. Tables 1 and 2 show that MOEA/D-HMPS performs significantly better than the other algorithms. According to Figure 7(d), the performances of MOEA/D-KF, PPS, and SGEA are greatly influenced by the jumping changes in F9, but not MOEA/D-HMPS. This is because MOEA/D-HMPS stores information about different jumps and the information can be reused when similar jumps subsequently occur, which effectively mitigates the impact of the jumps. In comparison with the problems discussed above, in F10, the geometric shapes of two consecutive PFs are completely different. Tables 1 and 2 show that MOEA/D-KF has the best performance among these algorithms. MOEA/D-HMPS is not effective on F10 given that it adopts differential prediction based on the current

individuals; however, the subsequent environment is completely different from the current one, which results in prediction errors.

### 5.3. Influence of Change Severity and Frequency

The difficulty level of DMOPs is determined by the severity and frequency of change. To solve DMOPs having different difficulties, a DMOEA should be able to adapt the optimal solutions to the moving optima efficiently and effectively. To investigate the robustness of MOEA/D-HMPS further, we tested it on two representative problems with different combinations of change severity and frequency: FDA1 and dMOP2. In the experiments described in this subsection on the influence of the severity of change,  $\tau_t$  was fixed to 30 and  $n_t$  to 5, 10, and 20, respectively. Similarly, in the experiments on the influence of change frequency,  $n_t$  was fixed to 10 and  $\tau_t$  to 10, 20, and 30, respectively. The box-plots of MIGD values, shown in Figures 8 and 9, were obtained for MOEA/D-HMPS and the best algorithm among the three comparative algorithms (see Table 1) on each test problem. On FDA1 and dMOP2, MOEA/D-KF is the best among the three comparative algorithms.

#### 5.3.1. Influence of change severity

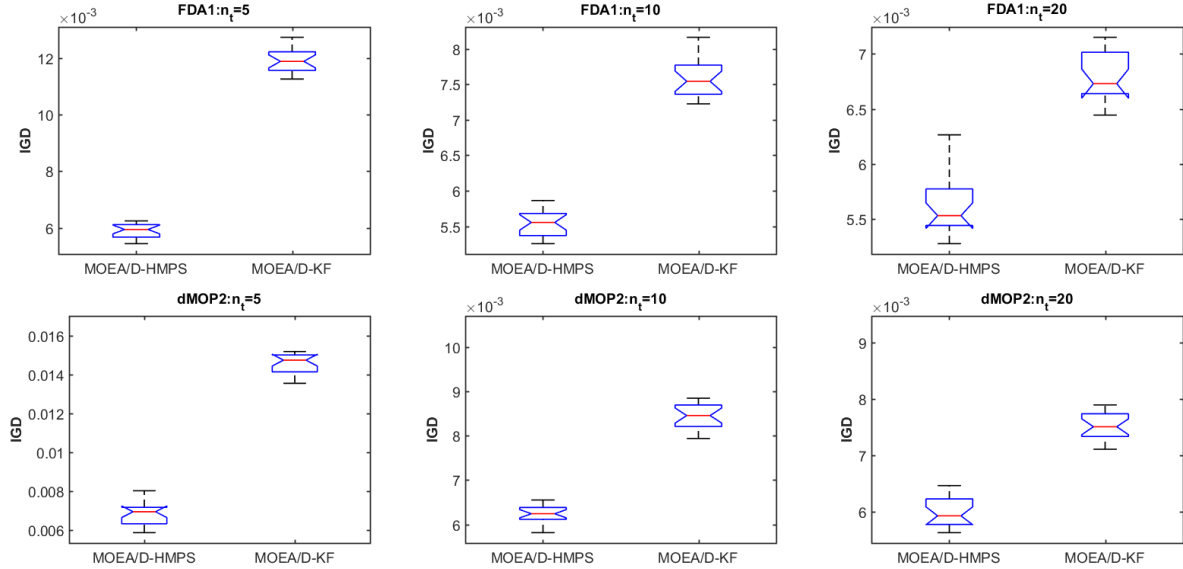


Figure 8: Influence of the severity of change on FDA1 and dMOP2 problems. The frequency of change  $\tau_t$  was set to 30. The figures show the box plot of inverted generational distance values for the multiobjective evolutionary algorithm based on decomposition with a hybrid of memory and prediction strategies (MOEA/D-HMPS) and the best algorithm among the three comparative algorithms for the two benchmark problems for  $n_t = 5, 10$ , and 20. Each row shows a particular benchmark problem and the value of  $n_t$  varies from 5 to 20.

In Figure 8, it is observed that, as  $n_t$  increases, i.e., the severity of change reduces, the performance of both MOEA/D-HMPS and MOEA/D-KF improves, whereas, as the severity increases, the performance of MOEA/D-KF is affected more easily than that of MOEA/D-HMPS. MOEA/D-HMPS is more robust to various severities of change.

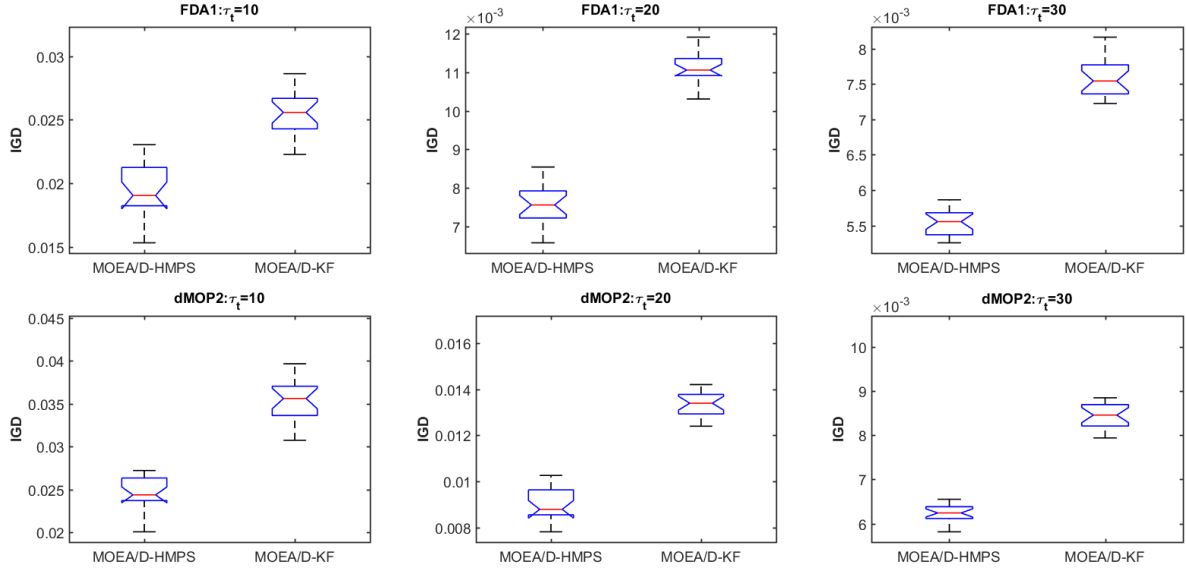


Figure 9: Influence of the frequency of change on FDA1 and dMOP2 problems. The severity of change  $n_t$  was set to 10. The figures show the box plot of inverted generational distance values for the multiobjective evolutionary algorithm based on decomposition with a hybrid of memory and prediction strategies (MOEA/D-HMPS) and the best algorithm among the three comparative algorithms for the three benchmark problems for  $\tau_t = 10, 20$ , and  $30$ . Each row shows a particular benchmark problem and the value of  $\tau_t$  varies from 10 to 30.

### 5.3.2. Influence of change frequency

In Figure 9, MOEA/D-HMPS is also shown to perform better than the comparative algorithms on the three benchmark problems with different change frequencies. This observation suggests that MOEA/D-HMPS is able to adapt the solutions to the moving optima more efficiently. Although the  
540 underlying MOEAs of MOEA/D-HMPS and MOEA/D-KF are the same, the tracking mechanism of MOEA/D-HMPS is more effective.

### 5.4. Effects of the Key Components of Hybrid of Memory and Prediction Strategies

This subsection is devoted to examining the contributions of the memory and prediction strategies that are the key components of MOEA/D-HMPS. We conducted comparison experiments using two  
545 variants of HMPS. One adopts only the differential prediction strategy based on population center points, also called the center-point prediction strategy (CPS). The second employs only the memory strategy based on population center points, called the center-point memory strategy (CMS). In CMS, when we are dealing with a dissimilar change, the 50% top ranked existing individuals are reused, and the remaining individuals are randomly reinitialized. CPS and CMS are also incorporated in MOEA/D;  
550 the resultant algorithms are denoted by MOEA/D-CPS and MOEA/D-CMS, respectively. We compared MOEA/D-CPS, MOEA/D-CMS, and MOEA/D-HMPS on all test problems. The experimental results are shown in Table 3.

The MIGD values in Table 3 show that MOEA/D-HMPS performs significantly better than MOEA/D-CPS and MOEA/D-CMS on most of the test instances, which demonstrates the efficiency of HMPS  
555 and that the two strategies are indispensable.

Table 3: Mean and standard deviation values of modified inverted generational distance metric obtained by two variants and multiobjective evolutionary algorithm based on decomposition with a hybrid of memory and prediction strategies (MOEA/D-HMPS)

Problem	MOEA/D-CPS	MOEA/D-CMS	MOEA/D-HMPS
FDA1	7.752E-3(2.348E-4)(-)	7.911E-3(2.850E-4)(-)	<b>5.621E-3(2.062E-4)</b>
FDA2	5.645E-3(4.370E-4)(-)	5.461E-3(3.132E-4)( $\approx$ )	<b>5.423E-3(5.875E-4)</b>
FDA3	1.821E-2(6.417E-4)(-)	1.721E-2(1.394E-3)(-)	<b>1.559E-2(9.470E-4)</b>
FDA4	8.585E-2(4.190E-3)(-)	8.449E-2(2.874E-3)(-)	<b>8.150E-2(2.502E-3)</b>
FDA5	1.865E-1(2.458E-2)(-)	1.870E-1(3.334E-2)(-)	<b>1.439E-1(1.297E-2)</b>
dMOP1	8.049E-3(1.342E-3)( $\approx$ )	<b>7.526E-3(1.083E-3)(<math>\approx</math>)</b>	8.054E-3(1.036E-3)
dMOP2	8.683E-3(4.628E-4)(-)	9.368E-3(4.713E-4)(-)	<b>6.374E-3(3.713E-4)</b>
dMOP3	2.456E-2(1.783E-3)(-)	3.112E-2(1.967E-3)(-)	<b>2.320E-2(2.058E-3)</b>
F5	2.817E-2(2.626E-3)(-)	5.487E-2(1.279E-2)(-)	<b>2.122E-2(5.577E-3)</b>
F6	5.717E-2(1.110E-2)( $\approx$ )	7.211E-2(1.740E-2)(-)	<b>5.324E-2(9.546E-3)</b>
F7	3.238E-2(7.864E-3)( $\approx$ )	3.830E-2(7.304E-3)(-)	<b>2.926E-2(6.961E-3)</b>
F8	<b>1.015E-1(1.562E-3)(+)</b>	1.147E-1(3.820E-3)(-)	1.058E-1(1.935E-3)
F9	4.775E-2(5.560E-3)(-)	5.662E-2(1.984E-2)(-)	<b>2.138E-2(2.496E-3)</b>
F10	1.057E-1(8.968E-3)(-)	2.328E-1(2.596E-2)(-)	<b>8.137E-2(8.934E-3)</b>

In MOEA/D-HMPS, the prediction strategy follows the same direction as the DSS proposed in [40]. It is interesting to compare the performances of MOEA/D-HMPS and DSS. As the experimental results presented in the **Supplementary Materials** show, MOEA/D-HMPS performs significantly better than DSS on most of the test instances, which also demonstrates that HMPS is effective for solving DMOPs.

We also conducted experiments to study the influence of storing different historical information, as shown in the **Supplementary Materials**. Much of the historical information extracted from the current optimal population can be archived, such as the population center, the center point and variance of the population distribution, and the entire population. The comparison results presented in the **Supplementary Materials** show that the strategy applied in MOEA/D-HMPS, i.e., storing the population centers, obtains a better compromise than the other two strategies.

### 5.5. Influence of Multiobjective Optimizers

The underlying MOEA plays a vital role in solving DMOPs. In this section, we discuss experiments on two typical MOEAs, MOEA/D and NSGA-II, combined with HMPS, and compare the performance of the two resultant algorithms on the test problems. To allow a fair comparison, in the experiments both MOEA/D and NSGA-II employed a differential evolution (DE) crossover operator. Seven test instances with different PF changes were used to assess the two combinations, MOEA/D-HMPS and NSGA-II-HMPS. The experimental results are shown in Table 4.



Table 4: Mean and standard deviation values of modified inverted generational distance metric obtained by non-dominated sorting genetic algorithm-II with a hybrid of memory and prediction strategies (NSGA-II-HMPS) and multi-objective evolutionary algorithm based on decomposition with a hybrid of memory and prediction strategies (MOEA/D-HMPS)

Problem	NSGA-II-HMPS	MOEA/D-HMPS
FDA1	2.671E-2(5.219E-3)(-)	<b>5.621E-3(2.062E-4)</b>
FDA3	6.484E-2(1.763E-2)(-)	<b>1.559E-2(9.470E-4)</b>
FDA4	3.192E-1(9.065E-3)(-)	<b>8.150E-2(2.502E-3)</b>
FDA5	3.625E-1(1.203E-2)(-)	<b>1.439E-1(1.297E-2)</b>
dMOP2	5.044E-2(9.224E-3)(-)	<b>6.374E-3(3.713E-4)</b>
F9	4.617E-2(6.014E-3)(-)	<b>2.138E-2(2.496E-3)</b>
F10	<b>8.014E-2(7.452E-3)(<math>\approx</math>)</b>	8.137E-2(8.934E-3)

As the MIGD values in Table 4 show, MOEA/D-HMPS performs significantly better than NSGA-II-HMPS on most of the test instances, which indicates that the underlying MOEA/D is more effective than NSGA-II in dealing with the FDA, dMOP, and F test problems. However, as indicated in [29], the distribution and the shape of the PF of some DMOPs may change dramatically, which poses significant challenges to MOEA/D. In this case, other MOEAs or an MOEA/D with adaptive reference vectors could better suit the problems.

### 5.6. Running Time

In this section, we examine the running time of the algorithms. Four algorithms were run independently on two biobjective problems (FDA1 and F5) and two triobjective problems (FDA4 and F8) with  $\tau_t = 5$ . The running times of the algorithms are reported in Table 5. It can be clearly observed that MOEA/D-KF and PPS are more time consuming than MOEA/D-HMPS and SGEA. The main reason for this is that MOEA/D-KF and PPS adopt relatively complex prediction models. PPS stores a series of historical information at continuous time steps and takes a relatively long time to train the AR prediction model. MOEA/D-KF adopts the time-consuming KF prediction model to predict the new location of optimal solutions. Both MOEA/D-HMPS and SGEA employ a more time-efficient prediction method to predict the new location of optimal solutions. SGEA takes more time than MOEA/D-HMPS because, in the stationary evolutionary process, SGEA performs farthest-first selection to preserve elite individuals, which takes more time than the selection strategy used in MOEA/D-HMPS.

### 5.7. Effects of the Number of Adoptions in the Two Response Mechanisms

The number of environmental changes affects the use of the two response mechanisms in MOEA/D-HMPS. In this section, the experimental results for four representative test problems (FDA1, FDA3, dMOP3, and F5) are employed to examine the number of times the two response mechanisms are used under different numbers of environmental changes (40, 80, 120). The statistic results are presented in Table 6. In Table 6, it can be seen observed that, as the number of environmental changes increases,

Table 5: Runtime comparison

Problem	PPS	SGEA	MOEA/D-KF	MOEA/D-HMPS
FDA1	1611.73(s)	152.827(s)	12137.47(s)	24.103(s)
FDA4	1566.55(s)	172.536(s)	12170.4(s)	30.536(s)
F5	1611.19(s)	158.437(s)	11660(s)	21.171(s)
F8	1645.15(s)	153.108(s)	11546.1(s)	31.378(s)

Table 6: Number of uses of two response mechanisms under different number of environmental changes

Problem	Change Type	40	80	120
FDA1	Similar	19	59	99
	Dissimilar	21	21	21
FDA3	Similar	19	59	99
	Dissimilar	21	21	21
dMOP3	Similar	3	8	16
	Dissimilar	37	72	104
F5	Similar	1	41	81
	Dissimilar	39	39	39

the number of similar changes in FDA1, FDA3, and FDA5 also increases, whereas dMOP3 always has few similar changes. By considering these results together with the MIGD and MHV results reported in Tables 1 and 2, it can be seen that MOEA/D-HMPS outperforms the other algorithms on FDA1, FDA3, and FDA5. MOEA/D-HMPS is very competitive in terms of solving DMOPs with many cyclic changes. MOEA/D-KF performs better than MOEA/D-HMPS on dMOP3, which indicates that the KF prediction approach is more accurate than the proposed differential prediction strategy in the case of dissimilar changes. Overall, MOEA/D-HMPS is a promising algorithm for handling DMOPs and the proposed memory-driven prediction strategy is effective for solving DMOPs with cyclic changes.

## 6. Conclusions and Future Work

In this paper, we proposed a hybrid method of memory and prediction strategies, called HMPS, to improve the performance of MOEAs for handling DMOPs. When a change occurs, first HMPS identifies whether the new change is similar to some historical changes, and then a frequently used differential prediction method and a novel memory-driven prediction method are employed to handle dissimilar and similar changes, respectively. HMPS is embedded in an MOEA/D framework, resulting in the MOEA/D-HMPS algorithm. MOEA/D-HMPS was compared with state-of-the-art DMOEAs on a number of DMOPs with different dynamic characteristics and difficulties. In the experimental results, MOEA/D-HMPS showed a superior performance on the majority of the test problems, which demonstrates that it is able to track a moving PS and/or PF effectively. The experimental results of our studies on the influences of problem parameters and the effects of the memory-driven prediction strategy also suggest the robustness and effectiveness of HMPS and MOEA/D-HMPS.

Although MOEA/D-HMPS showed an encouraging performance on the test instances, we also noticed its weakness on DMOPs with few similar changes (dMOP3) and DMOPs with two completely different consecutive PFs (F10). Moreover, MOEA/D-HMPS should be tested on more DMOPs with more complicated properties. Means of reusing the stored historical optimal information more efficiently need to be discovered in future work. The algorithm could also be improved by combining the proposed strategy with other effective algorithms, such as EDA, or other dynamic handling techniques, for example diversity maintenance. The source code of the proposed algorithm is written in C++ and available at <https://github.com/CIA-SZU/ZSX>.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant nos. 61871272, 61471246, and 61673331), Project of Department of Education of Guangdong Province (Grant no. 2016KTSCX121), Guangdong Foundation of Outstanding Young Teachers in Higher Education Institutions (Grant no. Yq2013141), Guangdong Special Support Program of Top-Notch Young Professionals (Grant no. 2014TQ01X273), and Shenzhen Scientific Research and Development Funding Program (Grant nos. JCYJ20170302154227954, JCGG20170414111229388, and ZYC201105170243A).

## References

### References

- [1] C.R.B. Azevedo, A.F.R. Arajo, Generalized immigration schemes for dynamic evolutionary multiobjective optimization, in: 2011 IEEE Congress on Evolutionary Computation, pp. 2033–2040.
- [2] S. Biswas, S. Das, S. Kundu, G.R. Patra, Utilizing time-linkage property in DOPs: An information sharing based artificial bee colony algorithm for tracking multiple optima in uncertain environments, *Soft Computing* 18 (2014) 1199–1212.
- [3] S. Biswas, S. Kundu, S. Das, Inducing niching behavior in differential evolution through local information sharing, *IEEE Transactions on Evolutionary Computation* 19 (2015) 246–263.
- [4] J. Branke, Memory enhanced evolutionary algorithms for changing optimization problems, in: 1999 Congress on Evolutionary Computation, pp. 1875–1882.
- [5] R. Chen, K. Li, X. Yao, Dynamic multiobjectives optimization with a changing number of objectives, *IEEE Transactions on Evolutionary Computation* 22 (2018) 157–171.
- [6] M. Cmara, J. Ortega, F.D. Toro, Performance measures for dynamic multi-objective optimization, in: 2009 International Work-Conference on Artificial Neural Networks, pp. 760–767.
- [7] C. Coello, G. Lamont, D. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer, 2007.

- [8] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, 2001.
- [9] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation* 6 (2002) 182–197.
- [10] K. Deb, N. Rao, S. Karthik, Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling, in: 2007 International Conference on Evolutionary Multi-Criterion Optimization, pp. 803–817.
- [11] M. Farina, K. Deb, P. Amato, Dynamic multiobjective optimization problems: test cases, approximations, and applications, *IEEE Transactions on Evolutionary Computation* 8 (2004) 425–442.
- [12] S.B. Gee, K.C. Tan, H.A. Abbass, A benchmark test suite for dynamic evolutionary multiobjective optimization, *IEEE Transactions on Cybernetics* 47 (2017) 461–472.
- [13] S.B. Gee, K.C. Tan, C. Alippi, Solving multiobjective optimization problems in unknown dynamic environments: An inverse modeling approach, *IEEE Transactions on Cybernetics* 47 (2017) 4223–4234.
- [14] C. Goh, K.C. Tan, A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization, *IEEE Transactions on Evolutionary Computation* 13 (2009) 103–127.
- [15] C.K. Goh, K.C. Tan, An investigation on noisy environments in evolutionary multiobjective optimization, *IEEE Transactions on Evolutionary Computation* 11 (2007) 354–381.
- [16] M. Greeff, A.P. Engelbrecht, Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation, in: 2008 IEEE Congress on Evolutionary Computation, pp. 2917–2924.
- [17] I. Hatzakis, D. Wallace, Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach, in: 2006 Conference on Genetic and Evolutionary Computation, pp. 1201–1208.
- [18] M. Jiang, Z. Huang, L. Qiu, W. Huang, G.G. Yen, Transfer learning-based dynamic multiobjective optimization algorithms, *IEEE Transactions on Evolutionary Computation* 22 (2018) 501–514.
- [19] S. Jiang, S. Yang, Evolutionary dynamic multiobjective optimization: Benchmarks and algorithm comparisons, *IEEE Transactions on Cybernetics* 47 (2017) 198–211.
- [20] S. Jiang, S. Yang, A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization, *IEEE Transactions on Evolutionary Computation* 21 (2017) 65–82.
- [21] W.T. Koo, K.G. Chi, K.C. Tan, A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment, *Memetic Computing* 2 (2010) 87–110.

- [22] H. Li, K. Deb, Challenges for evolutionary multiobjective optimization algorithms in solving variable-length problems, in: 2017 IEEE Congress on Evolutionary Computation, pp. 2217–2224.
- [23] H. Li, Q. Zhang, Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II, IEEE Transactions on Evolutionary Computation 13 (2009) 284–302.
- [24] Q. Lin, J. Chen, Z. Zhan, W. Chen, C.A. Coello, Y. Yin, C. Lin, J. Zhang, A hybrid evolutionary immune algorithm for multiobjective optimization problems, IEEE Transactions on Evolutionary Computation 20 (2016) 711–729.
- [25] R. Liu, Y. Chen, W. Ma, C. Mu, L. Jiao, A novel cooperative coevolutionary dynamic multi-objective optimization algorithm using a new predictive model, Soft Computing 18 (2014) 1913–1929.
- [26] A. Muruganantham, K.C. Tan, P. Vadakkepat, Evolutionary dynamic multiobjective optimization via kalman filter prediction., IEEE Transactions on Cybernetics 46 (2016) 2862–2873.
- [27] Z. Peng, J. Zheng, J. Zou, M. Liu, Novel prediction and memory strategies for dynamic multiobjective optimization, Soft Computing 19 (2015) 2633–2653.
- [28] H. Richter, Detecting change in dynamic fitness landscapes, in: 2009 IEEE Congress on Evolutionary Computation, pp. 1613–1620.
- [29] M. Rong, D. Gong, Y. Zhang, A multi-direction prediction approach for dynamic multi-objective optimization, IEEE Transactions on Cybernetics PP (2018) 1–13.
- [30] G. Ruan, G. Yu, J. Zheng, J. Zou, S. Yang, The effect of diversity maintenance on prediction in dynamic multi-objective optimization, Applied Soft Computing 58 (2017) 631 – 647.
- [31] S. Sahmoud, H.R. Topcuoglu, A memory-based NSGA-II algorithm for dynamic multi-objective optimization problems, in: 2016 European Conference on the Applications of Evolutionary Computation, pp. 296–310.
- [32] J.R. Schott, Fault tolerant design using single and multi-criteria genetic algorithms, Cellular Immunology 37 (1995) 113.
- [33] R. Shang, L. Jiao, Y. Ren, L. Li, L. Wang, Quantum immune clonal coevolutionary algorithm for dynamic multiobjective optimization, Soft Computing 18 (2014) 743–756.
- [34] E. Tantar, A. Tantar, P. Bouvry, On dynamic multi-objective optimization, classification and performance measures, in: 2011 IEEE Congress of Evolutionary Computation, pp. 2759–2766.
- [35] A. Trivedi, D. Srinivasan, K. Sanyal, A. Ghosh, A survey of multiobjective evolutionary algorithms based on decomposition, IEEE Transactions on Evolutionary Computation 21 (2017) 440–462.

- [36] L. Tseng, C. Chen, Multiple trajectory search for unconstrained/constrained multi-objective optimization, in: 2009 IEEE Congress on Evolutionary Computation, pp. 1951–1958.
- 715 [37] Y. Wang, B. Li, Investigation of memory-based multi-objective optimization evolutionary algorithm in dynamic environment, in: 2009 IEEE Congress on Evolutionary Computation, pp. 630–637.
- [38] J. Wei, Y. Wang, Hyper rectangle search based particle swarm algorithm for dynamic constrained multi-objective optimization problems, in: 2012 IEEE Congress on Evolutionary Computation, 720 pp. 1–8.
- [39] F. Wilcoxon, Individual comparisons by ranking methods, *Biometrics Bulletin* 1 (1945) 80–83.
- [40] Y. Wu, Y. Jin, X. Liu, A directed search strategy for evolutionary dynamic multiobjective optimization, *Soft Computing* 19 (2015) 3221–3235.
- [41] Y. Yuan, H. Xu, B. Wang, B. Zhang, X. Yao, Balancing convergence and diversity in 725 decomposition-based many-objective optimizers, *IEEE Transactions on Evolutionary Computation* 20 (2016) 180–198.
- [42] S. Zeng, R. Jiao, C. Li, X. Li, J.S. Alkasassbeh, A general framework of dynamic constrained multiobjective evolutionary algorithms for constrained optimization, *IEEE Transactions on Cybernetics* 47 (2017) 2678–2688.
- 730 [43] Z. Zhan, J. Li, J. Cao, J. Zhang, H.S. Chung, Y. Shi, Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems, *IEEE Transactions on Cybernetics* 43 (2013) 445–463.
- [44] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Transactions on Evolutionary Computation* 11 (2007) 712–731.
- 735 [45] Q. Zhang, A. Zhou, Y. Jin, RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm, *IEEE Transactions on Evolutionary Computation* 12 (2008) 41–63.
- [46] A. Zhou, Y. Jin, Q. Zhang, A population prediction strategy for evolutionary dynamic multiobjective optimization, *IEEE Transactions on Cybernetics* 44 (2014) 40–53.
- [47] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, E. Tsang, Prediction-based population re-initialization 740 for evolutionary dynamic multi-objective optimization, in: 2007 International Conference on Evolutionary Multi-Criterion Optimization, pp. 832–846.
- [48] J. Zou, Q. Li, S. Yang, H. Bai, J. Zheng, A prediction strategy based on center points and knee points for evolutionary dynamic multi-objective optimization, *Applied Soft Computing* 61 (2017) 806–818.

## Supplementary Materials

### 1. Study of Storing Different Historical Information

This section devotes to studying the effects of storing different historical optimal information. There are three storage strategies in the experimental study, including: 1) Store the entire population as an archive element. 2) Store the centroid point and centroid variance of the population distribution as an archive element. 3) Store the population center as an archive element. In responding to a similar change, the first memory strategy reuses entire population to replace the existing population, the second memory strategy generates new individuals by using Gaussian distribution, where the stored centroid point and centroid variance are the average value and variance, and the third memory strategy is adopted in the proposed algorithm, which employs a novel memory-driven prediction strategy to relocate the optimal solutions. Three strategies are implemented with MOEA/D, denoted as MOEA/D-HMPS-S1 (use the first memory strategy), MOEA/D-HMPS-S2 (use the second memory strategy), MOEA/D-HMPS (use the third memory strategy). Some representative test instances are adopted to assess the performance of the three algorithms, the experimental results are shown in Table 1 and Table 2, and Figure 1 shows the box plot of IGD values obtained by MOEA/D-HMPS and MOEA/D-HMPS-S1 on some test instances.

Table 1: Experimental results of MOEA/D-HMPS-S1 and MOEA/D-HMPS

Problems	MOEA/D-HMPS-S1	MOEA/D-HMPS
FDA1	5.725E-3(2.629E-4)( $\approx$ )	<b>5.621E-3(2.062E-4)</b>
FDA3	<b>1.322E-2(1.442E-3)(+)</b>	1.559E-2(9.470E-4)
FDA4	<b>7.700E-2(3.621E-3)(+)</b>	8.150E-2(2.502E-3)
FDA5	1.945E-1(9.738E-2)( $\approx$ )	<b>1.439E-1(1.297E-2)</b>
dMOP2	<b>6.148E-3(2.353E-4)(+)</b>	6.374E-3(3.713E-4)
F5	2.233E-2(6.259E-3)( $\approx$ )	<b>2.122E-2(5.577E-3)</b>
F8	<b>9.808E-2(3.119E-3)(+)</b>	1.058E-1(1.935E-3)
F9	2.724E-2(1.378E-2)( $\approx$ )	<b>2.138E-2(2.496E-3)</b>
F10	9.820E-2(1.279E-2)(-)	<b>8.137E-2(8.934E-3)</b>

+/-/ $\approx$  indicates that the performance of the corresponding algorithm is significantly better than, significantly worse than, and comparable to that of MOEA/D-HMPS, respectively, at the 5% significance level under Wilcoxon rank-sum test.

From the statistical results of Table 1, it is clear to see that, MOEA/D-HMPS-S1 performs better than MOEA/D-HMPS on FDA3, FDA4, dMOP2 and F8, while on FDA1, FDA5, F5 and F9, they perform comparably, which indicates that using the first memory strategy is also very effective in handling DMOPs. However, from Figure 1 and the standard deviation in Table 1, it can be observed that, on FDA1, FDA5, F5, F9 and F10, MOEA/D-HMPS performs more stable than MOEA/D-HMPS-S1. According to the results of Table 2, MOEA/D-HMPS performs significantly better than

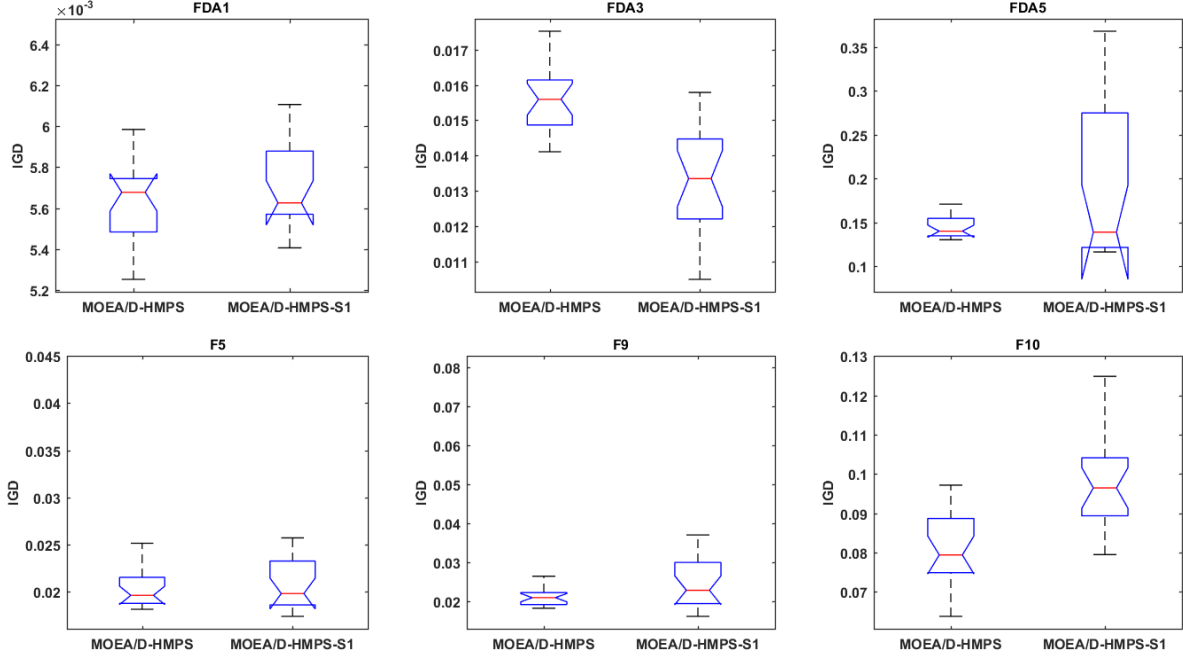


Figure 1: The box plot of IGD values obtained by MOEA/D-HMPS and MOEA/D-HMPS-S1 on some test instances.

Table 2: Experimental results of MOEA/D-HMPS-S2 and MOEA/D-HMPS

Problems	MOEA/D-HMPS-S2	MOEA/D-HMPS
FDA1	6.378E-3(1.301E-4)(-)	<b>5.621E-3(2.062E-4)</b>
FDA3	2.025E-2(1.480E-3)(-)	<b>1.559E-2(9.470E-4)</b>
FDA4	8.598E-2(2.993E-3)(-)	<b>8.150E-2(2.502E-3)</b>
FDA5	<b>1.337E-1(7.171E-3)(+)</b>	1.439E-1(1.297E-2)
dMOP2	6.868E-3(5.665E-4)(-)	<b>6.374E-3(3.713E-4)</b>
F5	6.119E-2(2.069E-2)(-)	<b>2.122E-2(5.577E-3)</b>
F8	1.113E-1(2.846E-3)(-)	<b>1.058E-1(1.935E-3)</b>
F9	7.305E-2(2.011E-2)(-)	<b>2.138E-2(2.496E-3)</b>
F10	1.264E-1(1.005E-2)(-)	<b>8.137E-2(8.934E-3)</b>

MOEA/D-HMPS-S2 on most test instances, which demonstrates that the memory-driven prediction strategy is more efficient than the second memory strategy. Therefore, by taking into account the robustness and performance of the algorithm, we adopt the third memory strategy in the proposed algorithm.

## 2. Performance comparison of MOEA/D-HMPS and DSS

In MOEA/D-HMPS, the prediction strategy is along the similar line of the DSS proposed in [1], it is interesting to compare the performance of MOEA/D-HMPS and DSS, which can better reflect the effectiveness of the combination of memory and prediction strategies. The statistical results of MIGD values for all test instances are presented in Table 3.



From the statistical results of MIGD values in Table 3, it is shown that MOEA/D-HMPS performs significantly better than DSS on most of the test instances, which demonstrates that the hybrid strategy of memory and prediction techniques is effective in solving DMOPs. However, in the complicated F test suite, DSS outperforms MOEA/D-HMPS on F6, F7 and F10, which indicates that MOEA/D-HMPS still needs to improve when handling the DMOPs with sharp and irregular changes.

Table 3: Mean and Standard Deviation values of MIGD metric obtained by DSS and MOEA/D-HMPS

Problems	DSS	MOEA/D-HMPS
FDA1	9.367E-3(4.280E-4)(-)	<b>5.621E-3(2.062E-4)</b>
FDA2	1.019E-2(1.191E-3)(-)	<b>5.423E-3(5.875E-4)</b>
FDA3	<b>8.979E-3(6.546E-4)(+)</b>	1.559E-2(9.470E-4)
FDA4	1.023E-1(3.588E-3)(-)	<b>8.150E-2(2.502E-3)</b>
FDA5	1.592E-1(3.908E-3)(-)	<b>1.439E-1(1.297E-2)</b>
dMOP1	2.063E-2(3.683E-3)(-)	<b>8.054E-3(1.036E-3)</b>
dMOP2	1.057E-2(8.424E-4)(-)	<b>6.374E-3(3.713E-4)</b>
dMOP3	8.423E-2(2.190E-3)(+)	<b>2.320E-2(2.058E-3)</b>
F5	2.163E-2(1.547E-3)(-)	<b>2.122E-2(5.577E-3)</b>
F6	<b>2.761E-2(8.670E-3)(+)</b>	5.324E-2(9.546E-3)
F7	<b>2.164E-2(3.581E-3)(+)</b>	2.926E-2(6.961E-3)
F8	1.465E-1(2.609E-3)(-)	<b>1.058E-1(1.935E-3)</b>
F9	2.359E-2(1.742E-3)(-)	<b>2.138E-2(2.496E-3)</b>
F10	<b>3.103E-2(2.525E-3)(+)</b>	8.137E-2(8.934E-3)

+/-/ $\approx$  indicates that the performance of the corresponding algorithm is significantly better than, significantly worse than, and comparable to that of MOEA/D-HMPS, respectively, at the 5% significance level under Wilcoxon rank-sum test.

## References

- [1] Wu, Yan, Y. Jin, and X. Liu. “A directed search strategy for evolutionary dynamic multiobjective optimization.” Springer-Verlag, 2015.